

Designing AI for the Places the Cloud Does Not Reach

Most AI software assumes a fast link to a data center somewhere else. Offshore, underground, downrange, and inside accredited enclaves, that assumption quietly fails.

Dave Hoon
Chief Technology Officer, Norseman Defense Technologies · September 2026

The Assumption Nobody Writes Down

The dominant pattern in AI software puts a large model at the center and everything else around it. It works well, and it carries an assumption that rarely makes it into a requirements document: *there is always a fast, reliable path to a data center somewhere else.*

The problem is structural. Capable models live almost exclusively in the cloud, and when users downrange lose their fast link to that cloud, those models go dark. Worse, most software built on them either works fully or not at all. The application does not degrade gracefully. It simply stops being useful.

Defense doctrine has a term for this condition: **DDIL**, meaning denied, disrupted, intermittent, and limited. The environment is not unique to the battlefield. The same constraint applies to:

- a survey vessel
- an underground mine
- a rural clinic on a satellite link
- a disaster response team on a mesh radio
- a construction site with no cellular coverage

- a manufacturing cell with no outbound path
- a classified enclave with no internet connection

These are the users who most need decision support and who are served the least.

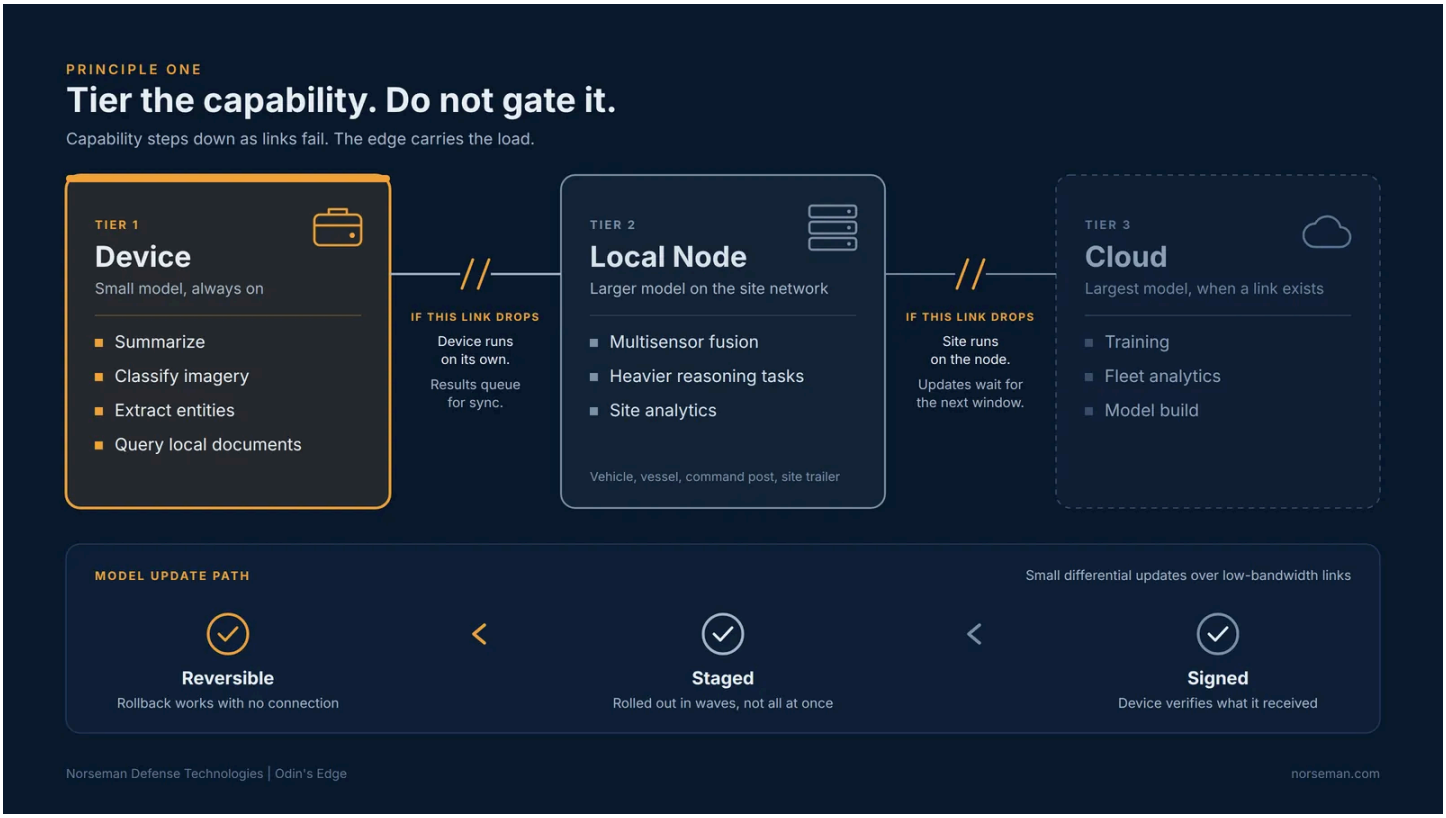


The light is on at the edge, not above it.

Principle One: Tier the Capability, Do Not Gate It

Design the application so capability degrades in steps rather than switching off. A practical tiering has three levels.

- **Device.** A small model runs locally and handles the routine, frequent work: summarizing, extracting entities, classifying imagery, and answering questions against a local document set.
- **Local node.** A larger model runs at a regional node, such as a vehicle, a vessel, a command post, or a site trailer, and devices reach it over a local network.
- **Cloud.** The largest model runs in the cloud and is used when a link exists and the task justifies it.



The user should never have to know which tier answered. **The application should know, and it should be honest about what it could not do at the current tier.**

Principle Two: Size the Model to the Task, Not to the Benchmark

The instinct to reach for the largest available model is expensive and often unnecessary. Purpose-built small language models have advanced quickly. On narrow, well defined mission tasks, models small enough to run air-gapped on a single device are closing much of the gap with far larger general models.

The underlying point holds across domains. *On a bounded task, a compressed model tuned for that task often beats a general model you cannot run at all.* Quantization and pruning are the standard techniques, and containerized microservices are the standard packaging.

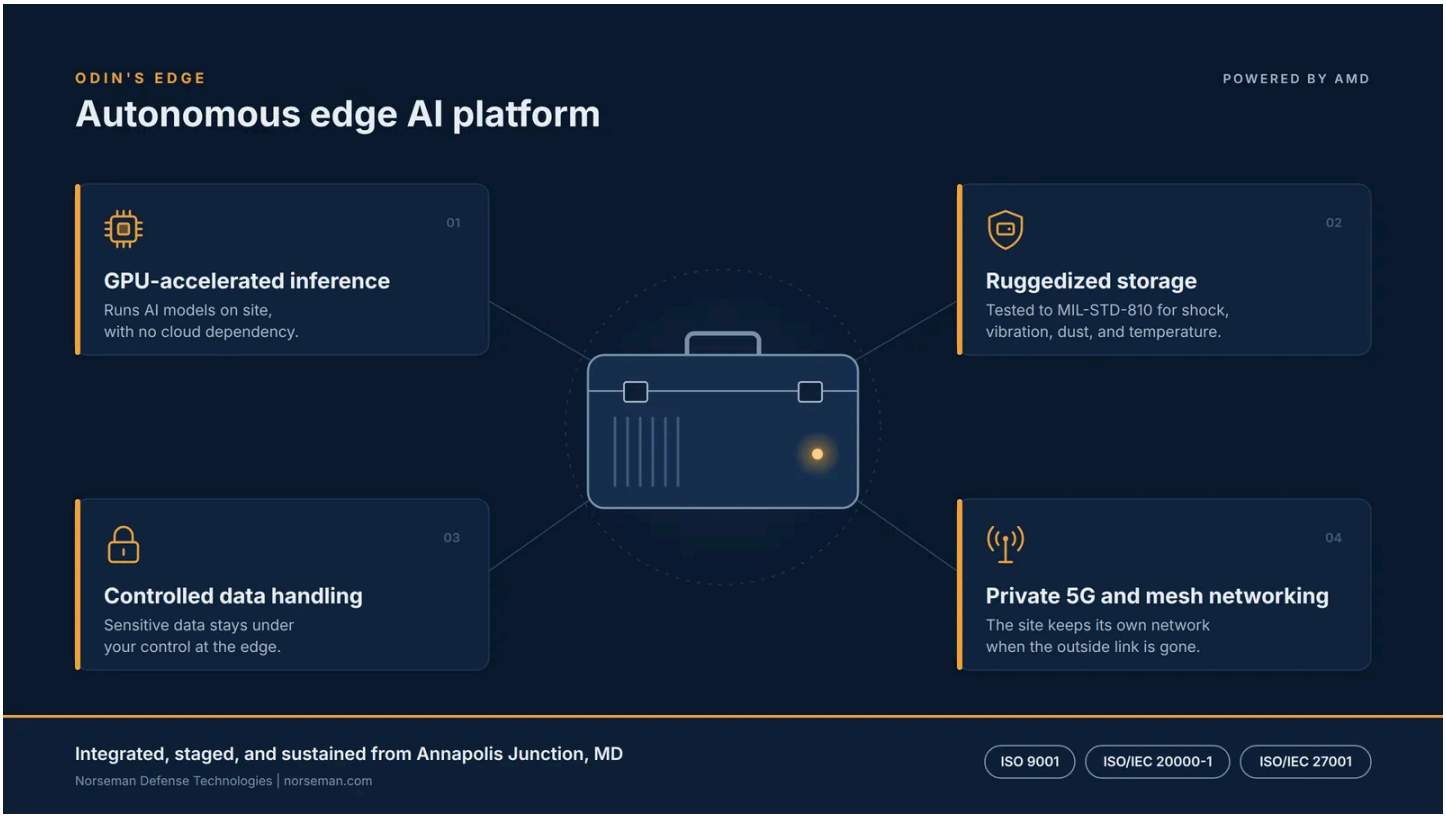
It follows that evaluation has to be local. A benchmark score earned on a hosted model says almost nothing about what the quantized version does on your hardware, with your data, under your latency budget. This is exactly the work that belongs in an integration lab before it belongs in a contract.

Principle Three: Respect the Hardware Envelope

Edge deployments are constrained by size, weight, and power (SWaP) before anything else. Hardware has to deliver maximum processing per watt, because an infantry team, a small autonomous craft, or a sensor on a remote pole cannot carry the battery a data center card assumes. Ruggedization matters for shock, dust, vibration, and temperature. Out-of-band management matters because you often cannot reach the operating system to fix something.

This is the design center for **Odin's Edge**, Norseman's autonomous edge AI platform. Each node includes:

- **GPU-accelerated inference** that runs AI models on site with no cloud dependency
- **Ruggedized storage** tested to MIL-STD-810 for shock, vibration, dust, and temperature
- **Controlled data handling**, so sensitive data stays under your control at the edge
- **Private 5G and mesh networking**, so a site keeps its own connectivity fabric when the outside link is gone



The nodes deploy into forward positions, vehicles, vessels, and site trailers, and they process everything locally with no dependence on the cloud.

Commercial and industrial deployments face similar constraints: thermal limits in an unconditioned cabinet, limited power on a remote pole, and the reality that nobody is driving four hours to reboot a box.

Principle Four: Treat Model Updates as Logistics

This is where programs succeed or fail, and it gets the least attention during selection.

A fielded model is a supply item. It has a version, an owner, an integrity requirement, and a delivery path that may be a low-bandwidth tactical radio or a weekly satellite window. To manage it, you need:

- **Differential updates** rather than full model pushes
- **Cryptographic signing**, so a device can verify what it received
- **Staged rollout**, so a bad model does not reach the whole fleet
- **A rollback that works with no connection**

The payoff is practical. When updates are small, signed, and staged, a new model can reach the fleet during a brief connectivity window instead of waiting for a maintenance visit, and a bad one can be pulled back the same way.

Logistics in the literal sense matters too. Someone has to stage, image, ship, track, and sustain the hardware. Odin's Edge nodes are built, integrated, and staged at Norseman's Annapolis Junction campus, which combines an integration lab, a warehouse, and a logistics hub under ISO-certified quality, service management, and information security programs. *That is the unglamorous half of edge AI, and it decides whether a pilot ever becomes a fleet.*

Principle Five: Plan the Reconnection

Disconnected operation produces data that has to reconcile when the link returns. Local inference results, operator corrections, newly collected data, and telemetry all need to sync without overwriting what happened elsewhere.

Decide in advance what wins in a conflict, what gets queued and what gets dropped, and how much local storage covers the longest realistic outage. Where the workflow has to reach an enterprise system of record, design that path in advance instead of improvising it. Odin's Edge integrates with ServiceNow, so work generated at the edge lands in the same operational workflow the rest of the organization already runs.

These are the questions that reveal whether a vendor has actually operated disconnected or only demonstrated it.

What to Ask a Vendor

1. What happens when the link drops in the middle of a task? Ask to watch it happen.
2. What is the smallest configuration you can deploy, and what capability does it lose?
3. How does a model update reach a device with 64 kbps of intermittent bandwidth?
4. How does the device verify what it received?
5. What is the sustained power draw under load, not at idle?
6. Can you show results on our task, on our hardware?
7. Who stages, ships, and sustains the fleet after the demo ends?

Make It Real

Building for disconnected operation does not mean a stripped-down version of the cloud architecture. **It is a different architecture, with different economics and a different definition of success.**

Norseman has spent three decades delivering infrastructure where the constraints are real. Odin's Edge exists because our customers kept running into the same wall: the capability was available everywhere except the place they needed it. We do the evaluation in our integration lab on the hardware you would actually field. We prove the disconnected behavior before it matters, and we stage and sustain the fleet from our own facility.

If your AI roadmap assumes connectivity you do not have, bring us the hardest site you operate. That is the one we want to see.

Explore **Odin's Edge**, our **Machine Learning** use cases, or **contact our team** to put AI where the cloud does not reach.

Published at norseman.com/insights/ai-disconnected-degraded-environments. To talk with Norseman Defense Technologies, call 410.579.8600 or email sales@norseman.com.